Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Currently amended) An overlapping command submitting ~~committing~~ method of dynamic cycle pipeline, for a chip having a pipeline ~~structure~~ including a plurality of stages, comprising the following steps:

(a) reading a [[the]] command from a command buffer and storing it in a command register;

(b) decoding the command;

(c) preprocessing operators of the command, preparing initial operators of each stage of the pipeline, and storing them into a initialization register;

(d) judging whether the pipeline is not full, if it is not full, directly inserting a new command ~~and then ending~~; otherwise, waiting for an exiting signal from the command in the pipeline in the last pipeline period before exiting;

(e) after receiving the exiting signal, judging whether there is command relevance between the new command to be inserted and an [[the]] old command to exit, if yes, then inserting the new command after the old command exit~~, and then ending~~; otherwise, performing a [[the]] next step;

(f) when the old command is in the last cycle of the pipeline, submitting ~~committing~~ the new command to the pipeline.

2. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein the Step (b) also includes a step of judging whether there is illegal command, if there is, then deleting the illegal command and returning to Step (a); otherwise, conducting the next step.

3. (Currently amended) The command submitting ~~committing~~ method of Claim 2, wherein said illegal command includes: the instructions with incorrect command code and/or carrying unreasonable command paramcters.

4. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein the exiting signal is released ~~before~~ two stages before ~~when~~ the new command enters the pipeline ~~stage~~.

5. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein the command relevance means that the new command and the old command cannot share the hardware processing module in the same [[one]] pipeline stage.

6. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein in the Step (e), it is also judged in which stage of the pipeline, a field ~~stage~~ switch shall be conducted for the new and old commands, and the field switch is completed in a [[the]] corresponding pipeline stage where the new and old commands overlaps.

7. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein in the Step (e), it is also judged whether there is any field conflict between the new command and the old command, if there is, then the field of the new command is added into the pipeline when submitting, and ~~committing, while~~ the field of the old command enters into the field branch and maintains in the field branch until the last time that the old command uses this field; in case there is no field conflict, a [[the]] field switch is conducted in a [[the] corresponding pipeline stage after submitting ~~committing~~.

8. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein in the Step (c), it is required to provide the initial status of ~~all kinds of~~ the commands at the entry to the pipeline.

9. (Currently amended) The command submitting ~~committing~~ method of Claim 1, wherein the said commands include reading/writing memory commands, reading/writing control register commands and various searching commands.

10. (Previously presented) A chip on which the method according to Claim 1, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

11. (Currently amended) The command submitting ~~committing~~ method of Claim 2, wherein the exiting signal is released ~~before~~ two stages before ~~when~~ the new command enters the pipeline ~~stage~~.

12. (Currently amended) The command submitting ~~committing~~ method of Claim 2 [[1]], wherein the command relevance means that the new command and the old command cannot share the hardware processing module in the same [[one]] pipeline stage.

13. (Previously presented) A chip on which the method according to Claim 2, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command

register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

14. (Previously presented) A chip on which the method according to Claim 3, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

15. (Previously presented) A chip on which the method according to Claim 6, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

16. (Previously presented) A chip on which the method according to Claim 7, is carried out having the cycle pipeline structure, comprising: interface of host computer,

input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

17. (Previously presented) A chip on which the method according to Claim 8, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.

18. (Previously presented) A chip on which the method according to Claim 9, is carried out having the cycle pipeline structure, comprising: interface of host computer, input buffer, command processing unit, and result unit; the command processing unit comprises: command interpreter and pipeline performing unit; characterized in that the command interpreter further comprises: command buffer controllers, command register, processing unit of operator, pipeline initialization register and control automaton, which are connected in order; the control automaton controls the command buffer controller to read a command from the command buffer, and stores the command into the command

register; the control automaton decodes the command, and controls the processing unit of operator to prepare initial operators of each pipeline stage according to the type of the command, and stores them into the pipeline initialization register.